AMYGDALA $z \mapsto z^2+c$

A Newsletter of Fractals & $\mathcal{M}$ (the Mandelbrot Set)
Issue #24
June 29, 1991

Copyright © 1986-1991, Rollo Silver

IN THIS ISSUE...

In *Structured Fractals From Random Transformations*, Roy Sommers discusses his program for creating fractals by using an iterated function system in which the transformations are chosen at random. The author takes that idea one step further by choosing the *coefficients* of the equations at random as well.

The author provides complete BASIC code for his program.

In Three IFS Programs for Macintosh, Gerald Edgar provides a concise discussion of iterated function systems, and then reviews in some detail three IFS programs written specifically for the Macintosh computer.

THE SLIDES (S24)

2319: **Raptor**, By Rollo Silver. A detail (relative magnification = 3.4) of JDJ1 (#2318: see slide set #22). A cruel figure, suggesting a bird of prey — perhaps a kind of abstract Pterodactyl. Against a violet surround, a figure covered with a brown skin, with a green interior forming an intriguing structural motif: a central cross bearing two hammerheads left and right. The structure is recursive: each hammerhead is itself the motif of cross with two hammerheads. The elements of the motif project rays which form the phalanges of eight brown talons grasping outward. Two talons from the cross, three each from the hammerheads. Center = −1.9963,7791,841 + 0.0000,1985,894i x 6.76x10^8, DL = 4095, ER = 2.5, PX = 1024x683, contrast CLUT.

2504: By Ian Entwistle.

2505: By Ian Entwistle.

3078: By Ken Philip.

STRUCTURED FRACTALS FROM RANDOM TRANSFORMATIONS

— *Roy W. Sommers*

M. Barnsley and A. Sloan have shown[1] that fractals and complex pictures composed of fractals can be generated from a relatively small number of specified affine transformations applied in random order. This allows pictures to be reproduced later from the stored coefficients of the transformations, thereby avoiding the storage of the thousands of pixel coordinates normally needed.

The transformations take the form:

```
XNEW =
 A(K)*XOLD + B(K)*YOLD + E(K)          [1]
YNEW =
 C(K)*XOLD + D(K)*YOLD + F(K)          [2]
```

where A(K), ... F(K) are the coefficients for the K^{th} transformation.

In generating a picture, the transformations are applied iteratively with XOLD and YOLD being replaced at each step by XNEW and YNEW respec-

---

[1] *Byte* (January 1988) p.215

CONTENTS

IN THIS ISSUE... 1
THE SLIDES (S24) 1
STRUCTURED FRACTALS FROM RANDOM TRANSFORMATIONS 1
THREE IFS PROGRAMS FOR MACINTOSH 3
AMYGDALO, AMYGDALAS, AMYGDALAT 9
SUBMITTING ARTICLES 9
AUTHOR 9
CIRCULATION/RENEWAL 9

tively, and the value of K (i.e. the transformation chosen for the given step) is selected with a probability proportional to $|A_K D_K - B_K C_K|$.

Barnsley and Sloan suggest that one method for determining the values of the coefficients required to reproduce a given picture is to use the Collage Theorem approach: to overlay the target picture with miniature versions of itself, and by regression analysis (or solving simultaneous equations) to determine which coefficients will transform the target picture into each of the overlaid miniatures.

In the accompanying computer program I have taken the random aspect of the above procedure one step further by choosing the coefficients of the equations at random, subject to size limitations. The transformations are then applied iteratively as described above. The motivation for this is to determine if this random set of steps will lead to a mere scattering of points, or if, because of the underlying structure of the equations, it will result in structured fractals. As can be seen by running the program, the latter is true, and the program produces an endless array of formations resembling vapor trails, trees, clouds, colliding galaxies, fireworks, etc. The results are especially beautiful if one has a color monitor.

The program is set for a maximum of eight transformations with the number used for a given case chosen randomly from the range 2...5. Each transformation is identified by a separate color (if available). If only black and white are available, the pictures are not as well-defined and the number of transformations should be limited to 2 or 3. After approximately 2500 points have been plotted, the program selects a new set of transformations. If it is desired to obtain the coefficients for a given picture, any key can be struck, and the coefficients will be typed out. The picture can then be reproduced at any later date by running the program, typing "1" at the first program query, and then typing the number of transformations and the values of A...F and P when requested.

I have written the program in TURBO BASIC for an EGA color system. However the program can easily be modified for a different BASIC or a different number of colors or transformations.

Editorial Comment:

In line 540 of the code below I have indicated

```
10      '
20      ' CHAOSRS APPLIES RANDOM TRANSFORMATIONS TO CREATE STRUCTURED FRACTALS
30      ' DESIGNED AND WRITTEN BY ROY SOMMERS — MARCH 1988
40      '
50      RANDOMIZE TIMER
60      DIM A(8),B(8),C(8),D(8),E(8),F(8),P(8),PS(8)
70      SCREEN 9                              ' PREFERABLE: USE SCREEN WITH AT LEAST 3 COLORS
80      WINDOW(-1.5,-1.5)-(1.5,1.5)           ' SCALE SCREEN TO X,Y OF -1.5, -1.5 TO 1.5, 1.5
50      ' NTR IS MAXIMUM NUMBER OF TRANSFORMATIONS (CAN BE SET FROM 2 TO 8)
60      INPUT "TYPE NUMBER OF COLORS AVAILABLE — NOT INCLUDING BACKGROUND ",CA
70      IF CA < 5 THEN NTR = CA + 1 ELSE NTR = 5
80      INPUT "TO REPRODUCE PREVIOUS PICTURE TYPE 1 OTHERWISE TYPE 2",PIC
90      M = INT((NTR-1)*RND+2)                'PICK RANDOM NUMBER OF TRANSFORMATIONS
100     I = 0: PT = 0                         'INITIALIZE
110     ' SELECT RANDOM COEFFICIENTS FOR TRANSFORMATIONS IN THE RANGE -.707 TO .707
120     ' AND PICK PROBABILITIES PROPORTIONAL TO ABS(A(K)*D(K) - B(K)*C(K))
130     IF PIC=1 THEN INPUT "TYPE NUMBER OF TRANSFORMATIONS ",M
140     FOR K=1 TO M
150     A(K)=RND*1.414-.707 : B(K)=RND*1.414-.707 : C(K)=RND*1.414-.707
160     D(K)=RND*1.414-.707 : E(K)=RND*1.414-.707 : F(K)=RND*1.414-.707
170     P(K)=ABS(A(K)*D(K)-B(K)*C(K))         'NUMERATOR OF PROBABILITIES
180     IF PIC<>1 THEN 200
190     INPUT "INPUT A,B,C,D,E,F,P ",A(K),B(K),C(K),D(K),E(K),F(K),P(K)
200     IF P(K)<.001 THEN P(K)=.01            'REJECT VERY SMALL PROBABILITIES
210     PT=PT+P(K)                            'SUM UP VALUES OF P(K)
220     PS(K)=PT                              'NUMERATOR OF CUMULATIVE PROBABILITIES
230     NEXT K
240     X=0 : Y=0 : XS=X : YS=Y               'STARTING VALUES FOR ITERATION
250     RANDOMIZE TIMER : CLS
260     ' BEGIN ITERATIVE LOOP
270     I=I+1
280     ' CHOOSE TRANSFORMATION TO BE USED FOR CURRENT POINT
```

```
290   PK=RND                                'CHOOSE RANDOM NUMBER FROM 0 TO 1
300   ' CHOOSE TRANSFORMATION TO BE USED FOR CURRENT POINT
310   J=0
320   ' LOOP TO CHOOSE TRANSFORMATIONS WITH A PROBABILITY OF PK(J)/PT
330   J=J+1
340   IF PK<=PS(J)/PT THEN 350 ELSE 330 ' COMPARE PK WITH CUMULATIVE PROB.
350   K=J
360   ' MAKE K'TH TRANSFORMATION
370   X=A(K)*XS+B(K)*YS+E(K)               ' CALCULATE NEW X IN TERMS OF OLD VALUES
380   Y=C(K)*XS+D(K)*YS+F(K)               ' CALCULATE NEW Y IN TERMS OF OLD VALUES
390   IF (X*X+Y*Y)>1000 THEN 90
400   CLOR=K                               ' SET COLOR EQUAL TO TRANSFORMATION NUMBER
410   IF CLOR>CA THEN CLOR=CA              ' RESTRICED BY AVAILABLE COLORS
420   IF I<10 THEN 440                     ' DO NOT PLOT FIRST 10 STEPS
430   PSET(X,Y),CLOR                       ' PLOT POINT AT X,Y WITH COLOR CLOR
440   XS=X : YS=Y                          ' SAVE OLD X AND Y AS XS AND YS
450   IF I<2500 THEN 270                   ' AFTER 2500 POINTS MOVE TO NEXT CASE
460   CLS                                  ' CLEAR SCREEN
470   ' IF YOU SEE SOMETHING YOU LIKE, HIT ANY KEY TO GET COEFFICIENTS (A-F),
480   ' PROBABILITIES (P) AND NUMBER OF TRANSFORMATIONS
490   PIC=2
500   IF INKEY$="" THEN 90                 ' IF KEY IS NOT HIT GET NEW TRANSFORMATIONS
510   CLS : SCREEN 0
520   PRINT "NUMBER OF TRANSFORMATIONS IS ",M : PRINT
530   FOR K=1 TO M
540   IF K=1 THEN PRINT "·····A·······B·······C·······D·······E·······F·······P"
550   PRINT USING "###.####";A(K),B(K),C(K),D(K),E(K),F(K),P(K)/PT
560   NEXT K
570   END
```

THREE IFS PROGRAMS FOR MACINTOSH

— *G. A. Edgar*

Iterated Function Systems

An *iterated function system*, or IFS, can be used to construct (and study) certain fractals. IFSs are used extensively in Barnsley's text *Fractals Everywhere*. They also appear, with less emphasis, in Falconer's text *Fractal Geometry: Mathematical Foundations and Applications* and my own text *Measure, Topology, and Fractal Geometry*.

Let us consider, as an illustration, the "Eiffel Tower" fractal pictured in Figure 1:

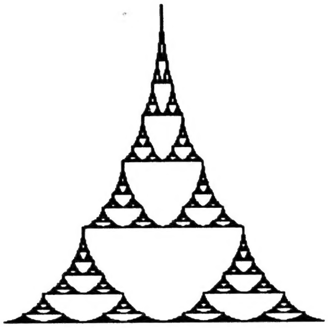


Figure 1— Eiffel Tower Fractal

The feature of this fractal that makes it ideal as an IFS is this: The picture is made up of three parts, or subimages, that look (almost) the same as the entire picture. To see this better, I can draw a box around the entire picture, then draw the corresponding box around each of the three subimages:

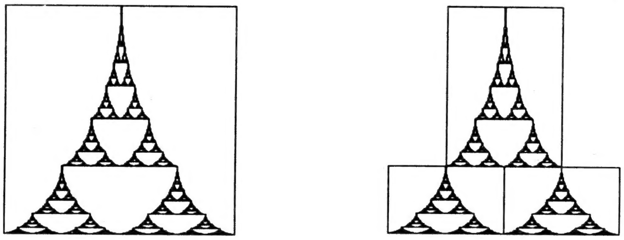


Figure 2— Decomposition of The Eiffel Tower

Of course, each of these three subimages is, in turn, made up of three more still smaller subimages. And so on.

The quantitative way to think of this is in terms of *affine transformations* of the plane. If we identify points of the plane with ordered pairs of real numbers, and identify these in turn with 2×1 column vectors, then an affine transformation f of the plane is a transformation defined in terms of a 2×2 matrix $\mathbf{A}$ and a "translation" vector u by the following formula:

$$f(x) = \mathbf{A}x + u$$

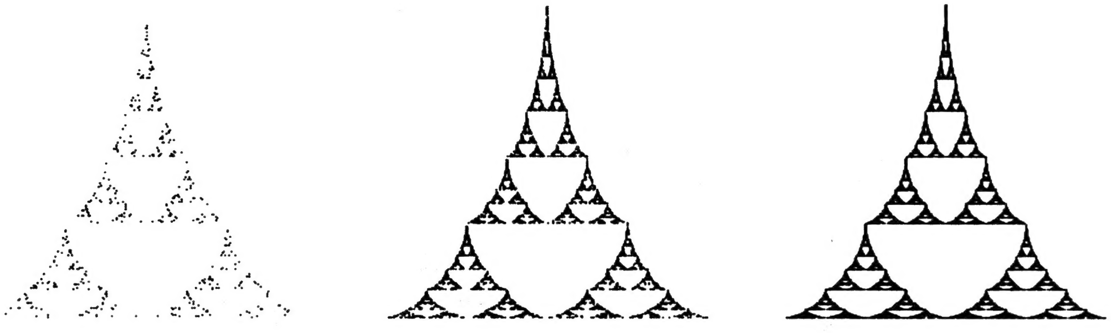


Figure 3— Eiffel Tower Comes Into Being

For the Eiffel Tower fractal, we use three transformations:

$$f_1\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$f_2\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

$$f_3\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.7 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0.25 \\ 0.3 \end{bmatrix}$$

The three functions taken together may be called an *iterated function system.* The fractal set in the plane that we want to construct should be a (compact nonempty) set K satisfying

$$K = f_1(K) \cup f_2(K) \cup f_3(K)$$

It can be proved that there is a unique set K satisfying these properties. It is called the *attractor* of the iterated function system (f_1, f_2, f_3).

There is more than one way to create a picture of the attractor for a given IFS. The most popular is "the chaos game", or the random method. Probabilities are assigned to the transformations. (For the Eiffel Tower, let's use probabilities 0.2, 0.3, 0.5 — they should be positive and add up to 1. Then we start with any point in the plane, say x_0. At each stage of the construction, after the point x_n has been generated, choose *at random* (according to the given probabilities) one of the transformations. Apply that transformation to the point x_n, the result will be a *new* point x_{n+1}. Continue in this way: for each step, make an independent random choice from among the transformations in the IFS and apply it to the point. It can be proved that (with probability 1) the sequence of points we get this way accumulates to the attractor. In practise, what this means is: if we ignore the first few points generated (the *transient* behavior), and then plot the remaining points, we will get a very good picture of the attractor. Figure 3, above, shows three pictures generated from the chaos game for the Eiffel Tower: the first has 500 points, the second 5000, and the third 50,000. At this resolution, It does not change much at all beyond 50,000.

Another construction for the attractor of an IFS is sometimes used. This is the *deterministic method.* We begin with any (compact nonempty) set in the plane, say A_0, and at each stage apply all of the transformations to all the points of the set A_n to get the new set A_{n+1}. In the example,

$$A_{n+1} = f_1(A_n) \cup f_2(A_n) \cup f_3(A_n)$$

This sequence of sets converges in an appropriate sense (the *Hausdorff distance*) to the attractor. On the computer, this time we begin with any (nonempty) bitmap for A_0, and at each stage apply the transformations to every point of the bitmap to get the new set. For example, if I start with a circle, I get the pictures shown in Figure 3, top of next page; the number of iterations in the pictures is: 0 (the circle itself), 1, 2, 3, 5, 10.

Figure 3 (next page) shows a few more sample fractals (one from each of the programs to be discussed); I can only show black and white pictures here. "Melissa" from the *Desktop Fractal Design System*, "Kangaroos" from *Fractal Attraction*, and "Spiral" from *IFS Explorer.*

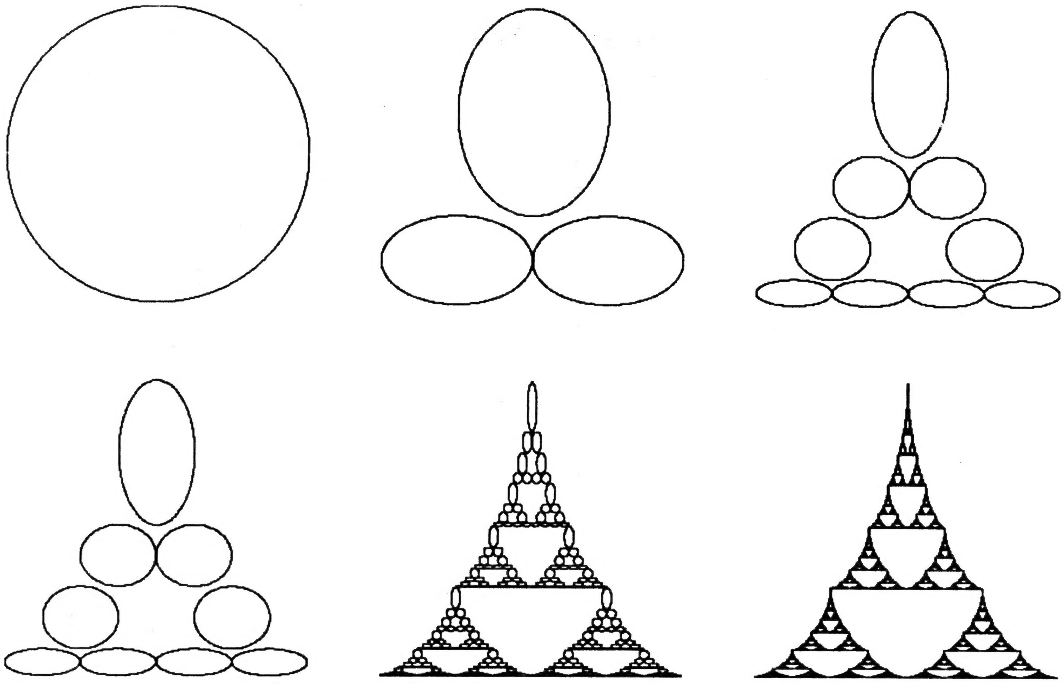


Figure 4— Eiffel Tower Made of Ovals

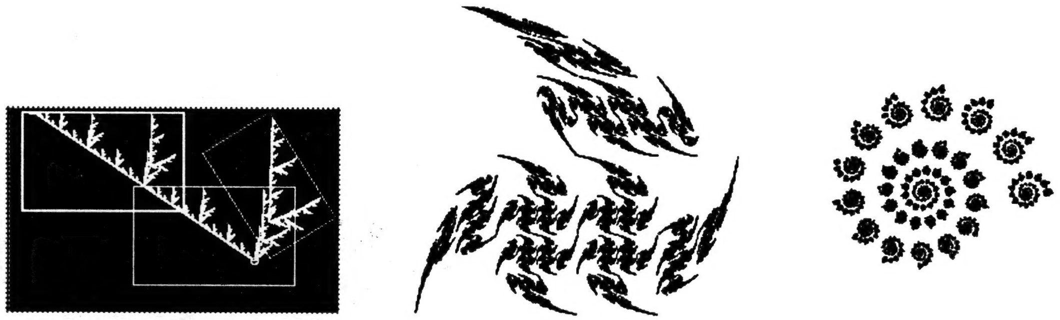


Figure 5— Sample Fractals

The Programs

Barnsley prepared a computer program for the use of students in his fractal geometry course at Georgia Tech. Students experienced in programming can write their own code for drawing pictures, but there is a need for a user-friendly program for non-programming students to use in exploring the IFS concept. Barnsley's program is called *The Desktop Fractal Design System.* There is a version for IBM-compatible computers, and a version for Macintosh. I used the Macintosh version for this review.

Two other IFS programs for Macintosh have been released lately, as well. *Fractal Attraction* is by K. Lee and Y. Cohen. *IFS Explorer* is by T. Koyn.

Each of the programs comes with a library of IFS codes that can be used to draw a variety of pictures. (The *real* fractal fanatic will have to buy all three of these programs so that she has a full set of all the images!) Other IFS codes are in the textbooks mentioned above. But of course the most interesting source is your own experimentation. Each of the programs comes with a booklet explaining not only the operation of the program, but also the basic ideas behind IFSs.

The basic mode of operation of each of the three programs is the same: The user enters a set of affine transformations of the plane by entering

Summary Table

	Desktop Fractal Design	Fractal Attraction	IFS Explorer
version reviewed	1.0	1.0	1.0.2
manual	40 pages	65 pages	72 pages
IFS library	27	58	70
RAM recommended	275K	1024K	384K
chaos game	yes	yes	yes
deterministic algorithm	no	yes	no
color rendering	transf	transf, dwell	none
color (max bits per pixel)	8?	24	1
save IFS code	TEXT	TEXT	special format
save fractal picture	no	PICT file, copy	PICT file, copy
graphic editing	good	excellent	none
Mac interface	fair	good	good

the coefficients (or reading them from a file). Then the program shows the corresponding attractor, using the "chaos game" algorithm. Each of the programs allows other variants, as well. For example, the deterministic algorithm; creating (or modifying) the transformations graphically with the mouse; specifying the transformations in scale/rotation form; magnification of a portion of the picture; coloring the picture according to the transformations; coloring the picture according to the "dwell".

A summary table of the features of the three programs is given above. Most of the items mentioned in the table are discussed more fully below.

Desktop Fractal Design System

This program is a professional-looking job. The chaos game happens *fast.* I was amazed by the speed (on a Mac IIci; and even on a Mac Plus). A nice feature allows you to zoom your picture up to the size of the entire screen.

In the "Library" screen, images of all the IFSs on the disk are shown at small size. One of them can be chosen from this gallery to be exhibited at large size or modified. This sure beats successively opening and processing each image by hand (or MacroMaker) in order to see the pictures, as you must do with the other two programs.

The *Desktop Fractal Design System* has three "Maps of Fractals". The map is a two-dimensional parameter space; each point in the map corresponds to a different fractal. Map 3 shows the familiar Mandelbrot set, where each point corresponds to a Julia set. Click on a point, and see the corresponding Julia set. Zoom in or out. Again, the speed amazed me. The maps shown in Figure 6 are lesser-known examples of the same idea, taken from Barnsley's book.

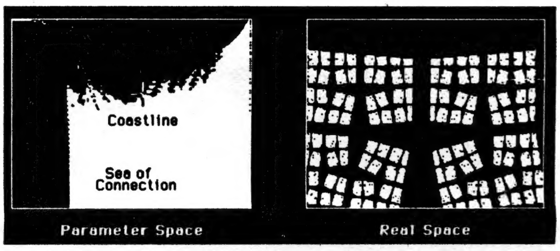


Figure 6— Parameter Space / Real Space

The program contains on-line help and informational screens. They are not as extensive as the manual, but they should answer questions about the program and inform the user about fractals.

I had some negative impressions of this program, too. Clearly, it is an IBM-type program that has been transplanted to the Macintosh. Apple's "human interface guidelines" are often not followed. These guidelines make all Mac programs feel similar, so that a user of the Mac already knows most of the basic features of every new program. But here you navigate from one screen to another by a system of buttons. Many of the screens fill the entire monitor, with no possibility to resize or move.

Cutting and pasting are not allowed: there is not even an *Edit* menu. So how is the user supposed to put her artistic results into other documents? I have an FKEY that lets me capture these pictures, but that is an un-Mac-like way to do it. The proper Mac way is to choose *Copy* from the Edit menu.

Graphical modification of an IFS with the mouse seemed clumsy to me. Making the modifications with buttons (*Shrink*, *Rotate*, *Squash*, *Expand*, etc.) was clearly the preferred method. In particular, I did not find a way to shear a transformation with the mouse. Changing the numerical coefficients for the transformations is done using up and down buttons, rather than just typing the desired value. IFS files are plain text, so they may be created, read, and modified, by any text editor.

The version I used (1.0) is apparently not compatible with (some) Mac II video cards. It bombed even with Apple's own "Macintosh Display Card 8*24". When I switched to the built-in video on my Mac IIci, it worked fine. And it worked fine on a Mac Plus. Hopefully, these incompatibilities can be easily remedied for future versions of the program.

Fractal Attraction

This program has three windows. One window shows the coefficients of the IFS. They may be entered and edited in the standard Mac way. A second window is the *Design* window; it shows the transformations graphically: this is done by showing an initial polygon (user definable) and its images under the transformations. In a color machine, each transformation is shown in its own color. Transformations may be modified with the mouse; this includes not only translations and resizing, but also rotations and shears. I found this editing environment the most intuitive of the three programs.

The last window is the *Fractal* window; it shows the picture of the fractal as it is generated. The "chaos game" random algorithm may be used, with a user-specified number of iterations. This is the only one of the three programs implementing the deterministic algorithm; again it may be iterated a user-specified number of times.

The fractal pictures may be copied to the Clipboard, or saved as PICT files. The IFS coefficients may be saved as well. When that is done, it is a plain text file, so that it can be created, edited, and examined by any text editor. I have used this feature to generate coefficients using another program, then processing them with *Fractal Attraction*. I did this to produce a sequence of images that show an animated fractal, gradually changing its form.

There are some facilities here for learning about the so-called "inverse problem": Given a picture, find an IFS that generates (a good approximation to) the picture. This idea has been in the news recently as a potentially useful way to compress the large amounts of data that are typically needed for digital storage or transmission of a picture. *Fractal Attraction* lets you experiment with this idea. It allows you to paste any picture into the Design window. Then, when defining your IFS transformations, you can trace parts of this picture for your initial polygon, see how the transformed images are related to the picture, and then make appropriate modifications in the transformations. Figure 7 shows an example from the manual. The original picture (a scanned maple leaf) is shown first, then an IFS attractor modeled on it:

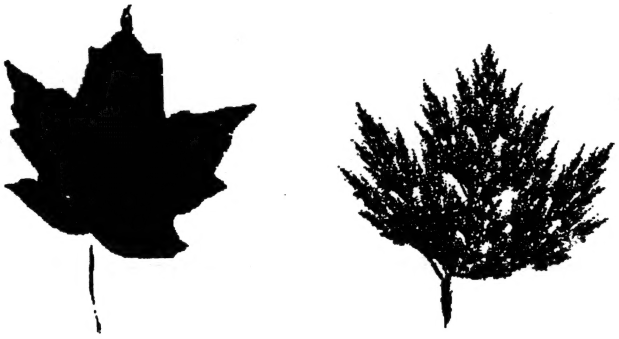


Figure 7— Maple Leaf and IFS Attractor

This IFS contains only 5 transformations; a reasonably realistic reproduction would take many more.

Color rendering (and grey-scale rendering) of the fractal can be done with this program. The color can be chosen based on the dwell. Pixels are colored according to the proportion of the time that they are hit. In conjunction with the considerations for the inverse problem, these techniques might be used as a compression method for *color* pictures.

You may have noticed in the summary table at the top of page 6 that I rated *Fractal Attraction* "good" on its Mac interface. A Mac user feels more at home when a program follows the interface guidelines, because the program behaves the way she unconsciously expects. I rated it only "good", rather than "excellent", because it still falls short in a few respects. Copy and Paste should be possible from anywhere to anywhere (as long as the data makes sense). I would like to be able to copy one transformation of an IFS and paste it back elsewhere in the list (either in the code window or in the design window). I would like to be able to paste a picture into the Fractal window, to use as the starting set in the deterministic algorithm. When I select a rectangle in the Fractal window for zooming, and I hold down the shift key at the same time, then the rectangle should be constrained to be a square.

IFS Explorer

Again, this program has most of the same IFS facilities as the others. There is a window in which the coordinates for the transformations are entered. They are entered as in a spreadsheet, rather than as in a word processor. The program will give you a lot of information on your transformations — probably a good feature when it is used for educational purposes.

There is a picture window where the image appears. The image in the window may be scrolled or magnified. *IFS Explorer* is particularly good on magnification, even millions of times. With the standard chaos game algorithm, when the picture is magnified this much, only a negligibly small proportion of the points generated in the chaos game fall within the window, so generating such a magnified picture takes a long time. *IFS Explorer* implements a "fast magnification algorithm" that may allow the picture to be shown within a reasonable time even when magnified millions of times. Figure 8 shows a "leaf" fractal, then a portion magnified by about 1000, and a portion magnified by about 1,000,000. A million-fold magnification done by the standard chaos game would not be practical, because of the time required.

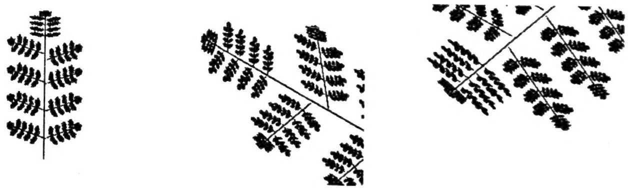


Figure 8— Leaf Fractal, Magnified

This program produces only black and white pictures. As with *Fractal Attraction, IFS Explorer* allows the fractal pictures generated to be copied to the Clipboard (and then moved to other Mac programs) or saved as a standard PICT file (which most Mac graphics programs can read). A long calculation may be run in the background under Multifinder while you do other things with the computer. This useful feature is missing in the other two programs.

IFS Explorer has no graphical editing of the transformations of the IFS itself. Saved IFSs are not standard text files, but special files that are useful only to this program. Data other than the coefficients of the transformations seems to be stored in the file.

I rated *IFS Explorer* "good" on its Mac interface. But, like *Fractal Attraction*, it has a few not-quite-right features. Often, even though it seems that something is wrong, you don't know quite what the problem is. For example, I got ready to copy a picture, pressed the *Command-C* key, and nothing happened. As noted above, when I select a rectangle in the picture window for zooming, and I hold down the shift key at the same time, then the rectangle should be constrained to be a square.

Data on the programs:

The Desktop Fractal Design System, Macintosh version, by M. Barnsley, from Academic Press (1250 Sixth Ave., San Diego, CA 92101), 1990, $39.95. ISBN # 0-12-079064-5

Fractal Attraction for the Macintosh, by K. Lee and Y. Cohen, from Sandpiper Software (P.O. Box 8012, St. Paul, MN 55108), 1990, $49.95

IFS Explorer, by T. Koyn, from Koyn Software (1754 Sprucedale, St. Louis, MO 63146), 1990, $49.95

AMYGDALO, AMYGDALAS, AMYGDALAT

More information about *"amygdala"* from the literature, courtesy of Hans Havermann:

FYI: from the "Dictionary of Protopharmacology", J. Worth Estes, Watson Publishing International, 1990.

The entry for **PERSICA** is "Flowers, seeds, and leaves of the peach, *Prunus persica*. Mild cathartic, sedative, and antihelminthic."

AMAYGDALAE, LAC: "Almond milk", oil from kernels of sweet or Jordan almonds, *Prunus amygdalus dulcis*, added to sugar water. Diluent; also used as an emollient and demulcent laxative, and to counteract CANTHARIS or other irritating medicines.

AMYGDALA, OL: Oil of bitter almonds, from *Prunus amygdalus amara* or, sometimes, from sweet almonds, *P.amygdalus dulcis*. Used internally as a sedative, antispasmodic, and expectorant, and externally as an emollient and to relax tense muscles. Said to be poisonous to dogs but not to men. Its poisonous principle, hydrocyanic acid, had been identified by Swedish chemist Karl Wilhelm Scheele in 1782, and was isolated from bitter almonds by German chemist Jeremiah Benjamin Richter in 1802.

AMYGDALUS COMMUNIS: Sweet almond; see **LAC AMAYGDALAE** and **OL AMAYGDALAE**.

AMYGDALUS PERSICA: Same as **PERSICA**.

SUBMITTING ARTICLES

Here are some guidelines for submitting articles for publication in Amygdala.

1. Type your article as you would like to see it appear in the newsletter. Please do not send handwritten drafts!

Send it (*my* order of preference): (A) On 3.5" diskette in Macintosh format for FrameMaker, MacWrite, WriteNow, or MS Word; or as a text file. Please also enclose paper copy so I can see your intent. (B) On 3.5" diskette in IBM format, text file. I have no way to deal with 5.25" diskettes. (C) Paper copy.

2. Illustrations should be either greyscale (suitable for halftoning) or black/white; not color! (A) Normally, illustrations will be printed in full column width, so you should make them 3.25" wide, if possible — provided that they're 300 dpi resolution. If they're grainier, make them larger if possible, so that they'll look good when reduced to 300 dpi. (B) Make sure that you clearly indicate which illustrations go where in the text! (C) All in all, it's better not to have captions welded into your pictures. Let me put them in ad lib. (D) I can handle illustrations on diskette in MacPaint, MacDraw, CricketDraw, or Adobe Illustrator formats.

3. Please send along a short biographical note, which I will try to publish in the same issue as your article.

4. Please include your telephone number, in case I have to reach you in a hurry with questions.

AUTHOR

G. A. Edgar (*Three IFS Programs*) received his Ph.D. at Harvard University in 1973. His research has been in measure theory, and its applications in functional analysis, probability, and fractal geometry. He is the author of the text, *Measure, Topology, and Fractal Geometry* (Springer-Verlag, 1990). He is currently on the faculty at The Ohio State University.

CIRCULATION

As of June 29, 1991 Amygdala has 600 subscribers, 210 of whom have the supplemental color slide subscription.

RENEWAL

For 96 of you subscribers out there this is the last issue of your current subscription. I urge you to use the enclosed form to renew your subscription promptly to avoid missing anything.

AMYGDALA, Box 219, San Cristobal, NM 87564
505/586-0197
See ORDER FORM for subscription information

MERZ

Fractal software, publications, and other stuff available from individuals and small companies.

Dada was a movement of artists and writers in the early 20th century who exploited accidental and incongruous effects in their work and who programmatically challenged established canons of art, thought and morality.

Dada was originally called *Merz* — the innermost syllable of *Commerzial.*

Merz is an issue-by-issue directory of products and services provided to our readers by individuals and small companies or organizations. We will accept ads for software of educational or recreational value, pamphlets, books, videotapes, clubs, organizations, T-shirts, fractal images, calendars, etc.

To place a Bulletin Board item, send camera-ready copy to fit into one or more rectangles — or fill out your copy by hand and we will typeset it. Be sure to include all relevant information, such as address and conditions of distribution. We will accept 1x1 areas (like top right), 1x2 areas (like this) 2x2 (like the one below) or 2x5 (full page).

Ads cost 30.00 per issue per basic rectangle. There is a 10% discount for 2 or more insertions placed at the same time (not necessarily for the same issue). Send to:

Amygdala Ads • Box 219 • San Cristobal, NM 87564
Telephone: 505/586-0197

FRACTAL CALENDAR — 1992

Seven original high-resolution fractal images adorn next year's Loyless Fractal Calendar. Includes some interesting color. Format is 11"x17" with large images.

FRACTAL ART PRINTS
Signed Limited Issues

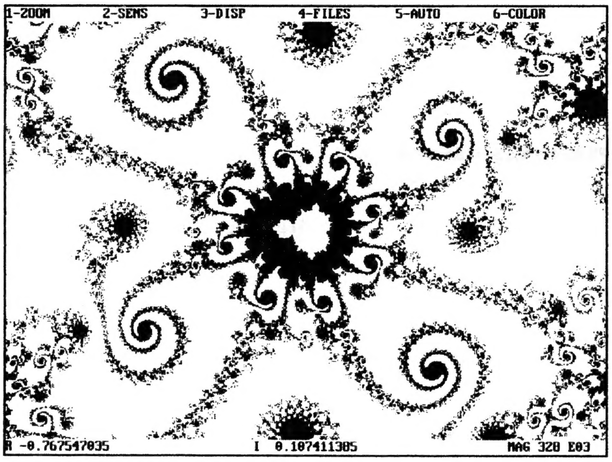
Prints are 11 inches square, numbered, titled, and signed by the artist: James Elliott Loyless. Shipped flat and shrink-wrapped for protection. Included with each is a detailed explanation of its creation, with formula and parameters. Resolution is 1440 x 1440 — over 2 million dots.

Please write for full product list and prices:

James E. Loyless
5185 Ashford Court
Lilburn, GA 30247

Mention Amygdala, Please!

If you place an order for a product as a result of seeing it advertised here, please let the advertiser know!



Minimum requirements:

- 80286 or 80386 Microprocessor
- MS DOS® 3.0 or later
- EGA, VGA or Super VGA graphics
- 640K RAM
- Hard Disk

ANDROMEDA RESEARCH, Inc.

Presents the Ultimate Fractal Software for the IBM® PC and Compatibles.

Fractal FX 2.0

- Hand optimized fixed point arithmetic and an efficient edge detection algorithm make this one of the fastest fractal programs in existence. No expensive math chip is needed.
- An extensive menu system makes Fractal FX easy to use.
- TIFF and GIF graphics files can be created for publication.
- We support most Super-VGA cards, even 1024 x 768 x 256.
- Fractal FX can magnify to 10^{72} where most programs are limited to 10^{12}.

Send $39.95 US to:
Andromeda Research, Inc.
6441 Enterprise Lane
Madison, WI 53719

or call toll-free:
1-800-755-3021
VISA or Mastercard accepted
Free Shipping!